

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-236286

(43)公開日 平成6年(1994)8月23日

(51)Int.Cl.<sup>5</sup>

G 0 6 F 9/46

識別記号

序内整理番号

F I

技術表示箇所

3 4 0 A 8120-5B

3 1 2 8120-5B

審査請求 未請求 請求項の数1 O L (全 8 頁)

(21)出願番号

特願平5-21413

(22)出願日

平成5年(1993)2月9日

(71)出願人 000002369

セイコーエプソン株式会社

東京都新宿区西新宿2丁目4番1号

(72)発明者 田辺 滋

長野県諏訪市大和3丁目3番5号 セイコ  
ーエプソン株式会社内

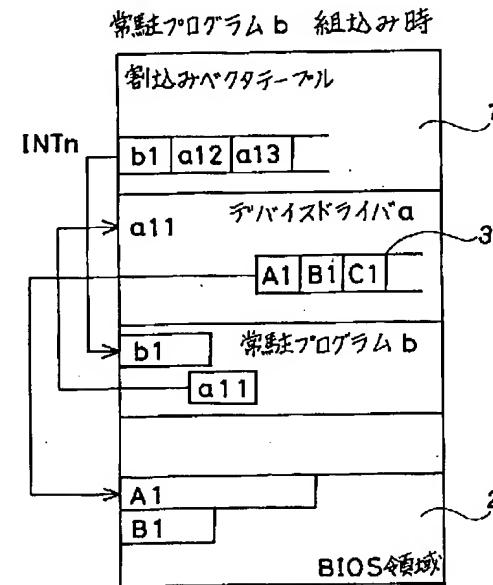
(74)代理人 弁理士 鈴木 喜三郎 (外1名)

(54)【発明の名称】 情報処理装置

(57)【要約】

【目的】 2つ以上の動作モードを動的に切り換えてそれぞれの動作モード上でアプリケーションを動作させることのできる情報処理装置において、動作モードの切り換えを意識しないデバイスドライバや常駐プログラムを正常動作させることを目的とする。

【構成】 マシンが起動した直後に、動作モードの切り換えにより内容が変化する割り込みベクタをトラップ処理する。動作モードが切り換わるごとにデータの受け渡しを行い、トラップした内容を常に正しいアドレスに書き換える。トラップした割り込みが発生したときに書き換えた正しいアドレスに処理を分岐する。



## 【特許請求の範囲】

【請求項1】割り込みベクタテーブルを用いて割り込み処理を行う第1のOSが起動した後、前記のOS上で第2のOSが起動し、又第2のOSの上で2つ以上の動作モードを切り換えてそれぞれの動作モード上でアプリケーションプログラムを動作させることのできる情報処理装置において、

第1のOSが起動した直後に任意の割り込みベクタテーブルをトラップする手段と、トラップした割り込みベクタテーブルを格納するテーブルと、

第2のOS上のアプリケーションプログラムの動作モードが変更されると、前記テーブルの内容を変更後の動作モードに対応した割り込みベクタテーブルに書き換える手段と、を有することを特徴とする情報処理装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、複数の動作モードを起動後に切り換えることのできる情報処理装置に関するものである。

## 【0002】

【従来の技術】パーソナルコンピュータのユーザーは、オペレーティングシステム（以下OSとする）を立ち上げた後、様々なデバイスドライバや常駐プログラムを組み込んでアプリケーションをより快適な環境で動作させるようにしている。例えばさらに広いメモリを得るためにメモリドライバやかな漢字入力を行うためのモジュールがこれに該当する。常駐プログラムを常駐させた場合のメモリマップを図8を用いて説明する。なお、図は本願の説明に必要な部分のみ概念的に示しており、一部省略している。パーソナルコンピュータが立ち上がると、BIOSプログラムによりシステム領域81、BIOS領域84、割り込みベクタテーブル85が割り付けられ、OSがOS領域83にロードされる。残りの部分はユーザーエリア82と呼ばれ、この中にアプリケーションプログラム領域86、常駐プログラム領域87がそのプログラムの実行に必要な分確保される。割り込みベクタテーブル85にはプログラムの「飛び先」を記述したテーブルがあり、このテーブルによってBIOS領域84内の処理ルーチンが呼び出され、BIOS処理やハードウェア割り込み等の処理が行われる。常駐プログラムは主メモリ上に常に存在し、必要に応じて割り込みベクタテーブル85の内容を割り込みベクタ格納領域88に格納した後、割り込みベクタテーブル85に自身をポイントするアドレスを設定することにより（この操作をトランプという）、本来の処理に替わって、または本来の処理の前に独自の処理を行う。

【0003】ところで、パーソナルコンピュータのような小型で個人向けの計算機システムでは從来から単一なOS、単一な動作モードで起動しアプリケーションを動作そして終了することが通常であった。このためデバ

イスドライバ、常駐プログラムはある固定の環境下でのみ正常動作するように作成されていた。その一方で最近、第1のOSの上で第2のOSが立ち上がりこの第2のOSの管理下で仮想的な第1のOSが複数存在する仮想計算機システムをサポートするOSが登場してきた。この新しいOSでは複数の従来の旧いOSのアプリケーションをそれぞれ異なった動作モードで実行することができるものがある。

【0004】複数の動作モードの切り換えについてより

10 具体的な例で説明すると、例えばユーザーの指定により表示画面の解像度をシステムのリセットを行うことなしに変えることができれば大変に便利である。なぜならば最近までは国内のパーソナルコンピュータでは640ドット×400ドットという比較的解像度の低い表示画面を持ったものが広く普及していた。ところがマンマシンインターフェースの向上や表示する情報量が増えるというメリットから、より高解像度の表示画面を持ったものがでてきたが、これに対応したアプリケーションの数は少ないからである。これらの2つのモードはVRAMの大きさが異なるためメモリーの配置を変える必要があるがハードは使用中に解像度を切り換えることができてもメモリ配置やBIOSが変わるので旧いOSではこれらの切り換えを動的に行うことができず、第2のOSでの制御が必要になる。

## 【0005】

【発明が解決しようとする課題】ところが、新しいOSが立ち上がる前に起動されメモリに常駐しているデバイスドライバやプログラムについては新しいOSで完全に制御することはむずかしく動作モードが途中で切り換わることに対応できずに固定の動作モードのまま処理を行うため、正常動作しなくなってしまう問題が発生する。このため新しいOSを立ち上げるときはこのようなデバイスドライバ、常駐プログラムは使用できないように制限をつけるか、または複数の動作モードで正常動作できるようおのおのデバイスドライバ、常駐プログラムを手直しなければならなかつた。

【0006】図9は動作モード1で第1のOSが立ち上がった直後を示している。INTnの割り込みベクタテーブル91nの内容はBIOS領域90の中の処理ルーチンA1をポイントしており、INTnによりA1が実行される。

【0007】図10は前記とは異なる動作モード2で第1のOSが立ち上がった直後を示しており、割り込みベクタテーブル91に動作モード1と異なる値が設定されているのが分かる。動作モード2ではINTnで処理ルーチンA2が実行される。

【0008】次に図11を用いて常駐プログラムの動作を説明する。パーソナルコンピュータが図9の状態で立ち上がった後、常駐プログラムbが起動されINTnをトラップする。具体的にはINTnの割り込みベク

3

ターブル91nの内容は常駐プログラムbのエントリーアドレス(処理ルーチンの先頭アドレス)bに書き換えられ、処理ルーチンA1のエントリーアドレスA1が常駐プログラムb内に格納される。以上の設定によりINTnで常駐プログラムbを介し処理ルーチンA1が実行される。

【0009】このように常駐プログラムbがメモリに常駐した後、INTnの割り込みが起こると常駐プログラムbに制御が移る。この後、常駐プログラムbの内部だけで処理が終了しINTnの呼び出し元にリターンする場合と、トラップする前のBIOSのエントリーアドレスA1に処理を戻す場合がある。常駐プログラムbが動作モード1で常駐した後動作モード2に切り換わり、トラップ前のBIOSへ処理を戻す場合に不具合が発生する。

【0010】この状態で動作モード2に切り換えた場合を図12を用いて説明する。動作モード2では割り込みベクターテーブル91及びBIOS領域90の内容が再配置され、一部を除き動作モード1との共通性はない。ここでは動作モード2のINTnの割り込みベクタアドレスは処理ルーチンA2に設定されている。ところが動作モード1で常駐した常駐プログラムbはINTnで処理ルーチンA1のエントリーアドレスをポイントしているが、処理ルーチンA1は存在せず、そのエントリーアドレスのポイントする処理ルーチンB2に制御が移り誤動作に至る。このため動作モードを切り換えるときに第2のOSでこのような常駐プログラムが内部領域に持っているトラップ前の割り込みベクターテーブルの内容を正しい値に書き直さないかぎりこれらの常駐プログラムやデバイスドライバを第2のOS以前に組み込むことはできなかった。

#### 【0011】

【課題を解決するための手段】本発明の情報処理装置は割り込みベクターテーブルを用いて割り込み処理を行う第1のOSが起動した後、前記のOS上で第2のOSが起動し、又第2のOSの上で2つ以上の動作モードを切り換えてそれぞれの動作モード上でアプリケーションプログラムを動作させることのできる情報処理装置において、第1のOSが起動した直後に任意の割り込みベクターテーブルをトラップする手段と、トラップした割り込みベクターテーブルを格納するテーブルと、第2のOS上のアプリケーションプログラムの動作モードが変更されると、前記テーブルの内容を変更後の動作モードに対応した割り込みベクターテーブルに書き換える手段と、を有することを特徴とし、動作モードの変化によって割り込みベクターテーブルの内容が変わる割り込みベクタ番号についてすべてトラップを行い内部に格納したトラップ以前の割り込みベクターテーブルの内容を第2のOS下で動作モードが変化するタイミングで正しい値に変更させるよう第2のOSとの間でデータのやり取りを行う割り込

4

みベクターテーブル制御のための専用デバイスドライバを第1のOSが立ち上がった直後に組み込むことにより、従来からのデバイスドライバ、常駐プログラムは動作モードによる割り込みベクターテーブルの内容の変化を受けなくなり正常に動作することができる。

#### 【0012】

【実施例】図1は第1のOSと第2のOSの関係、本発明により生み出された割り込みベクターテーブルをトラップするための割り込みベクターテーブル制御用デバイスドライバ(以下デバイスドライバ)aと第2のOSとの関係を概念的に図式化したものである。第1のOS上で常駐プログラム①、常駐プログラム②、デバイスドライバ①及び第2のOSが動作し、さらに第2のOS上で各動作モードのアプリケーションが動作している。第2のOS上のアプリケーションはデバイスドライバaを介して常駐プログラム、デバイスドライバにアクセスすることになる。

【0013】第1のOSが起動された直後の割り込みベクターテーブルとBIOS領域中の処理ルーチンは従来と同様に図9で示すようになる。ここでA1、B1……は割り込みベクタであり、動作モード1と動作モード2で値が変更される。(以後動作モードは2つで説明するが、それ以上であっても本質的に同様である)

この状態にデバイスドライバaを組み込んだものが図2である。第1のOSが立ち上がった直後にデバイスドライバaが起動されると、動作モードによって変更のある割り込みベクタをすべてトラップする。割り込みベクターテーブル1上のトラップ前の割り込みベクタA1、B1……はデバイスドライバa内のテーブル3に格納され、割り込みベクターテーブル1にはデバイスドライバa内の処理ルーチンをポイントする割り込みベクタa11、a12……が設定される。INTnが実行されるとデバイスドライバa内の処理ルーチンa11の制御によりテーブル3で設定されたA1がポイントするBIOS処理ルーチンA1が実行される。

【0014】割り込みベクタのトラップ処理の手順を図5を用いて説明する。デバイスドライバaは動作モード1と動作モード2の割り込みベクターテーブルを認識しており、従って動作モード1と動作モード2での割り込みベクタを変更すればよいかを知ることができる。デバイスドライバaが起動されるとトラップが必要な割り込みベクタアドレスを順次指定する(S1)。次に指定した割り込みベクタiの内容をデバイスドライバa内のメモリ(テーブル3)に格納する(S2)。割り込みベクタiにデバイスドライバa内の処理ルーチンa1iのエントリーアドレスa1iを設定する(S3)。上記をトラップが必要なすべて割り込みベクタについて行なったことを確認し(S4)、トラップ処理を終了する。

【0015】この後常駐プログラム(デバイスドライバでもよい)bがINTnの割り込みベクタをトラップ

したのを示したのが図3である。常駐プログラムbは割り込みベクタを自身のエントリーアドレスb1に換え、デバイスドライバaの処理ルーチンa11のアドレスを自身の内部メモリ領域に格納する。INTnにより常駐プログラムb、デバイスドライバaを経てBIOS処理ルーチンA1が実行される。

【0016】この状態で動作モード2に切り換わった場合、正常動作するためにはINTxxに対するBIOSのエントリーアドレスはA1がA2に、B1がB2にといったように切り換わらなければならない。第2のOSはオペレータの指示により動作モード変更を認識し、図6で示すように、デバイスドライバaに対して動作モード1の割り込みベクタを動作モード2の割り込みベクタに変更するよう指示する。ここで第2のOSはあらかじめ動作モード1、2の割り込みベクタテーブルの値がわかっており、ある特定の割り込みまたはデバイスコントロール等によりこれらの値を順にデバイスドライバaに通知することができる。

【0017】図4は動作モード2に切り換えた後のメモリの変移の概要を示す。また図7には、前述の処理によりデバイスドライバa内のテーブル3の割り込みベクタがモード2に対応した値に切り換わった状態を示す。図4の割り込みベクタa12、a13…はデバイスドライバaの処理ルーチンの先頭アドレスにすでに変更されているのでモードが変わっても不变である。ここでINTnが発生すると割り込みベクタテーブル1内の値b1により常駐プログラムbの処理ルーチンb1に制御が移る。さらにトラップ前の処理を行うときはテーブル3に格納されているa11の値によってデバイスドライバaの処理ルーチンa11に制御が移り、処理ルーチンa11はテーブル3の値A2によりBIOS処理ルーチンA2をポイントする。よってINTnにより動作モード2の正しい処理ルーチンA2を実行することができる。

【0018】このように見てきた時、デバイスドライバaを組み込んだ後にトラップを行うデバイスドライバや常駐プログラムは動作モードの切り換えによるBIOSのエントリーアドレスの変更から切り放され、動作モードに関わりなく動作することがわかる。

【0019】なお本発明の実施例はデバイスドライバaの形式で説明したが、常駐プログラムの形式であっても、一部制約はあるが同様の効果を得ることは可能である。

#### 【0020】

【発明の効果】以上述べた如く、本発明による複数の動作モードの切り換えによる割り込みベクタテーブルの変化を制御する専用のデバイスドライバをマシンの起動時に組み込むことにより、動作モードが固定であることを前提として作成された、割り込みベクタテーブルをト

ップする常駐プログラムやデバイスドライバ（ほぼ全数がこの形態）がこれらに修正を加えることなく、動作モードの切り換えをサポートするOSで利用できるので、従来の常駐プログラムやデバイスドライバを有効に活用できるとともに前記のOSの制御が簡便になるという効果を有する。

#### 【図面の簡単な説明】

【図1】第1のOSと第2のOSと割り込みベクタテーブル制御デバイスドライバとの関係を示す概念図。

10 【図2】図9の後、割り込みベクタテーブル制御デバイスドライバaが組み込まれたときの割り込みベクタテーブルの内容の変化を示した図。

【図3】図2の後、常駐プログラムbが組み込まれたときの割り込みベクタテーブルの内容の変化を示した図。

【図4】図3の後、動作モードが切り換わり、デバイスドライバaの内部メモリに格納したトラップ時の割り込みベクタテーブルの値が変わったことを示す図。

【図5】デバイスドライバaの初期化時に、必要な割り込みベクタのトラップ処理を行なうことを示すフローチャート。

20 【図6】デバイスドライバaの動作モード変更時、第2のOSによって呼ばれる処理フローチャート。

【図7】デバイスドライバaの内部メモリの割り込みベクタの格納領域が動作モードによって内容が書き換わったことを示した概念図。

【図8】常駐プログラムを示すメモリマップ。

【図9】動作モード1で第1のOS起動直後の割り込みベクタテーブルとBIOSとを示した図。

30 【図10】動作モード2で第1のOS起動直後の割り込みベクタテーブルとBIOSとを示した図。

【図11】図9の後、常駐プログラムbが常駐し、元の割り込みベクタテーブルの内容が常駐プログラムbの内部メモリに格納されたことを示す図。

【図12】図11の後、動作モードが2になり、BIOSの内容とそれにともなう割り込みベクタテーブルが変化することを示した図。

#### 【符号の説明】

1 : 割り込みベクタテーブル

2 : BIOS領域

40 3 : テーブル

81 : システム領域

82 : ユーザーエリア82

83 : OS領域

84 : BIOS領域

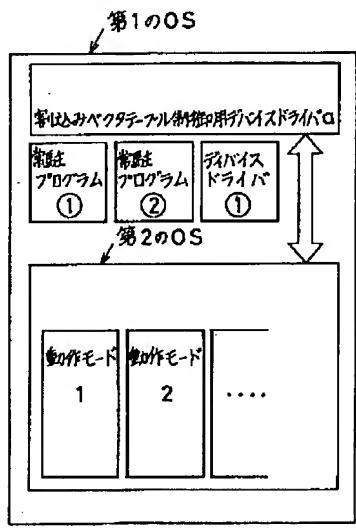
85 : 割り込みベクタテーブル

86 : アプリケーションプログラム領域

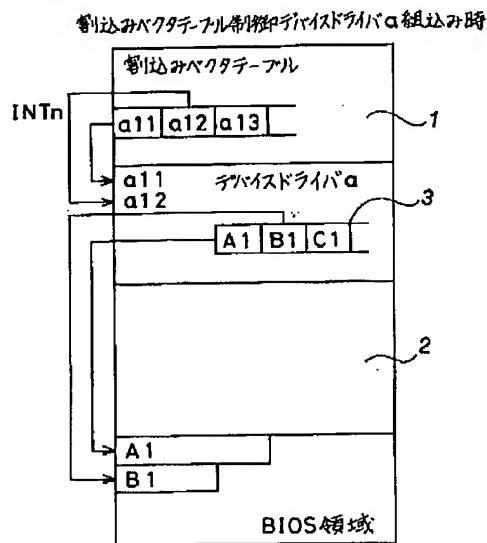
87 : 常駐プログラム領域

88 : 割り込みベクタ格納領域

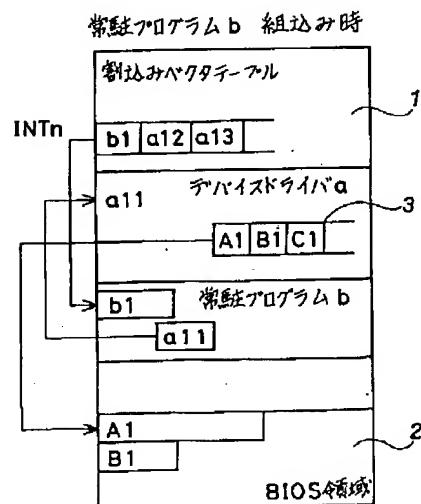
【図1】



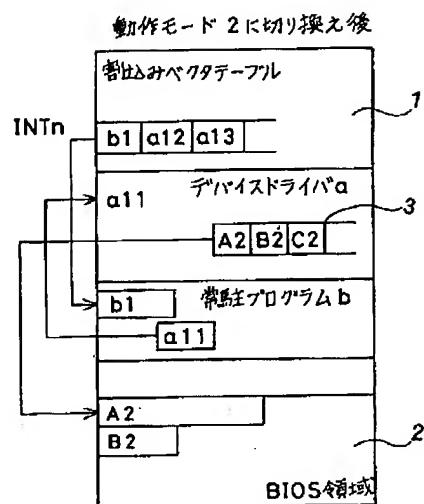
【図2】



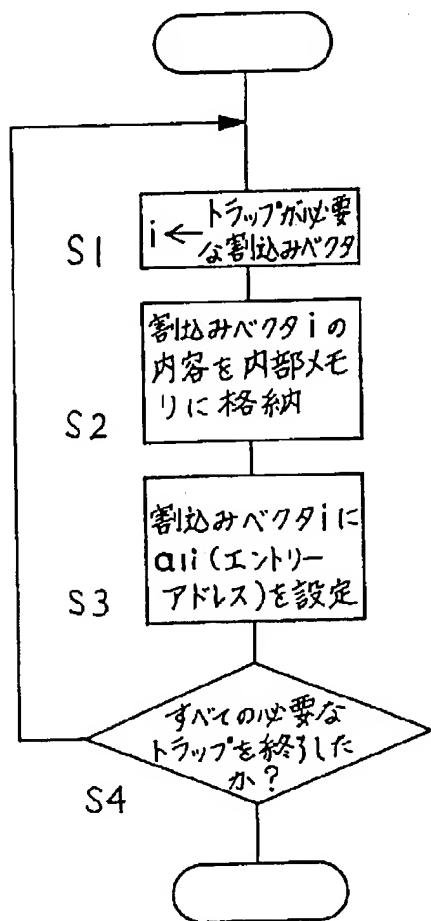
【図3】



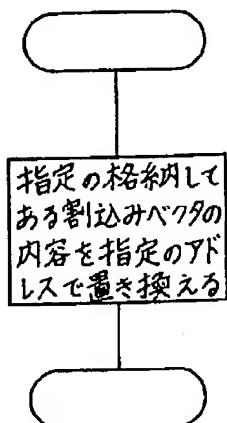
【図4】



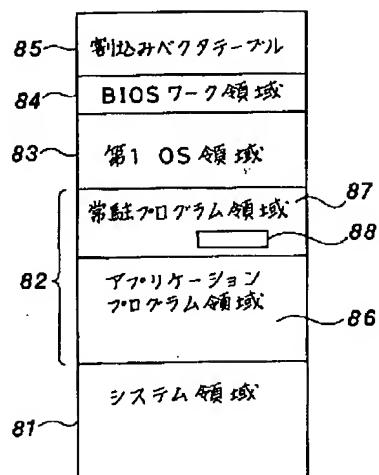
【図5】



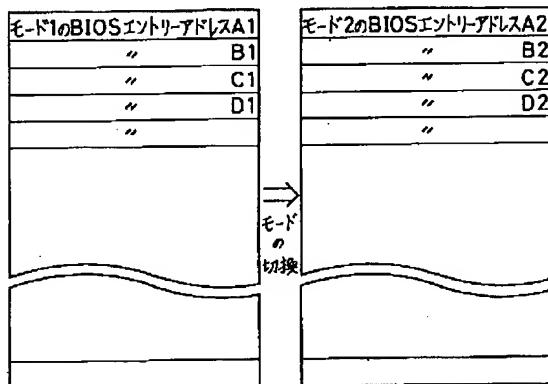
【図6】



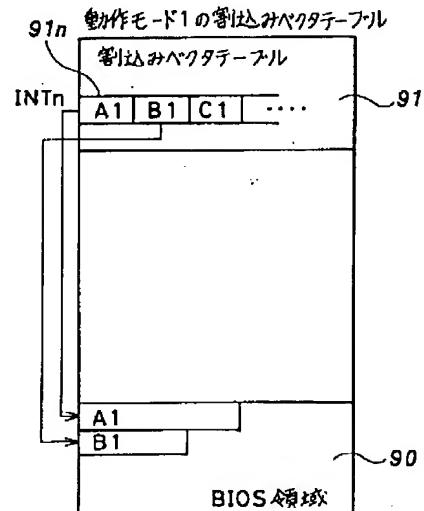
【図8】



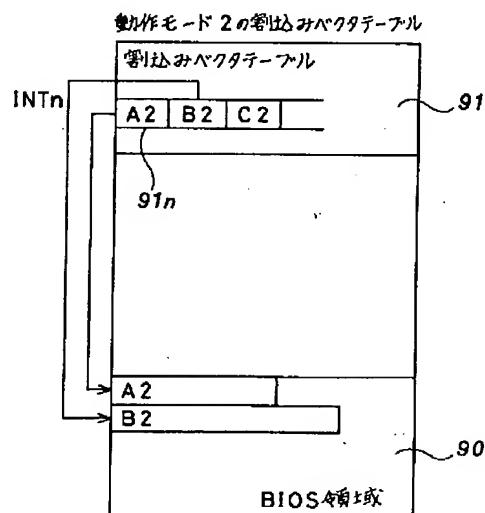
【図7】



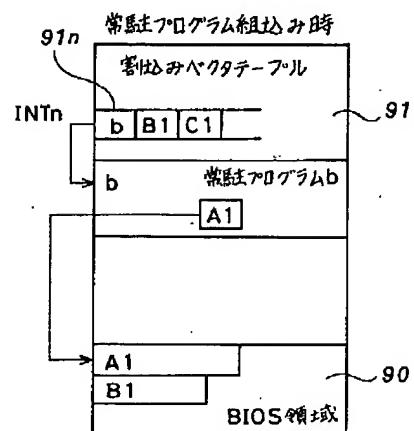
【図9】



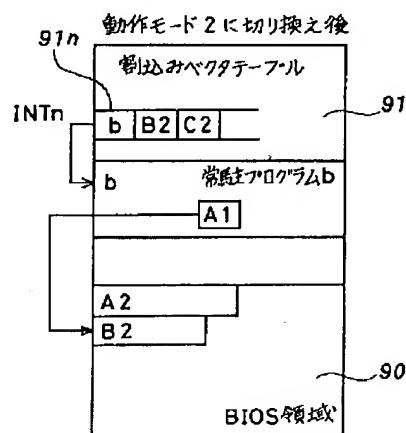
【図10】



【図11】



【図12】



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平6-236286

(43)公開日 平成6年(1994)8月23日

(51)Int.Cl.<sup>5</sup>  
G 0 6 F 9/46

識別記号 庁内整理番号  
3 4 0 A 8120-5B  
3 1 2 8120-5B

F I

技術表示箇所

審査請求 未請求 請求項の数1 O L (全 8 頁)

(21)出願番号

特願平5-21413

(22)出願日

平成5年(1993)2月9日

(71)出願人 000002369

セイコーエプソン株式会社

東京都新宿区西新宿2丁目4番1号

(72)発明者 田辺 滋

長野県諏訪市大和3丁目3番5号 セイコ  
ーエプソン株式会社内

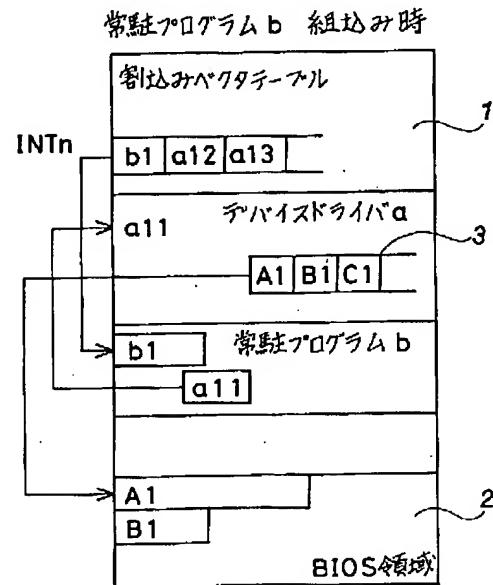
(74)代理人 弁理士 鈴木 喜三郎 (外1名)

(54)【発明の名称】 情報処理装置

(57)【要約】

【目的】 2つ以上の動作モードを動的に切り換えてそれぞれの動作モード上でアプリケーションを動作させることのできる情報処理装置において、動作モードの切り換えを意識しないデバイスドライバや常駐プログラムを正常動作させることを目的とする。

【構成】 マシンが起動した直後に、動作モードの切り換えにより内容が変化する割り込みベクタをトラップ処理する。動作モードが切り換わるごとにデータの受け渡しを行い、トラップした内容を常に正しいアドレスに書き換える。トラップした割り込みが発生したときに書き換えた正しいアドレスに処理を分岐する。



## 【特許請求の範囲】

【請求項1】 割り込みベクタテーブルを用いて割り込み処理を行う第1のOSが起動した後、前記のOS上で第2のOSが起動し、又第2のOSの上で2つ以上の動作モードを切り換えてそれぞれの動作モード上でアプリケーションプログラムを動作させることのできる情報処理装置において、

第1のOSが起動した直後に任意の割り込みベクタテーブルをトラップする手段と、トラップした割り込みベクタテーブルを格納するテーブルと、  
第2のOS上のアプリケーションプログラムの動作モードが変更されると、前記テーブルの内容を変更後の動作モードに対応した割り込みベクタテーブルに書き換える手段と、を有することを特徴とする情報処理装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、複数の動作モードを起動後に切り換えることのできる情報処理装置に関するものである。

## 【0002】

【従来の技術】パーソナルコンピュータのユーザーは、オペレーティングシステム（以下OSとする）を立ち上げた後、様々なデバイスドライバや常駐プログラムを組み込んでアプリケーションをより快適な環境で動作させるようにしている。例えばさらに広いメモリを得るためにメモリドライバやかな漢字入力を行うためのモジュールがこれに該当する。常駐プログラムを常駐させた場合のメモリマップを図8を用いて説明する。なお、図は本願の説明に必要な部分のみ概念的に示しており、一部省略している。パーソナルコンピュータが立ち上がると、BIOSプログラムによりシステム領域81、BIOS領域84、割り込みベクタテーブル85が割り付けられ、OSがOS領域83にロードされる。残りの部分はユーザーエリア82と呼ばれ、この中にアプリケーションプログラム領域86、常駐プログラム領域87がそのプログラムの実行に必要な分確保される。割り込みベクタテーブル85にはプログラムの「飛び先」を記述したテーブルがあり、このテーブルによってBIOS領域84内の処理ルーチンが呼び出され、BIOS処理やハードウエア割り込み等の処理が行われる。常駐プログラムは主メモリ上に常に存在し、必要に応じて割り込みベクタテーブル85の内容を割り込みベクタ格納領域88に格納した後、割り込みベクタテーブル85に自身をポイントするアドレスを設定することにより（この操作をトランプという）、本来の処理に換わって、または本来の処理の前に独自の処理を行う。

【0003】ところで、パーソナルコンピュータのような小型で個人向けの計算機システムでは従来から単一なOS、単一な動作モードで起動しアプリケーションを動作しそして終了することが通常であった。このためデバ

イストライバ、常駐プログラムはある固定の環境下でのみ正常動作するように作成されていた。その一方で最近、第1のOSの上で第2のOSが立ち上がりこの第2のOSの管理下で仮想的な第1のOSが複数存在する仮想計算機システムをサポートするOSが登場してきた。この新しいOSでは複数の従来の旧いOSのアプリケーションをそれぞれ異なった動作モードで実行することができるものがある。

【0004】複数の動作モードの切り換えについてより

- 10 具体的な例で説明すると、例えばユーザの指定により表示画面の解像度をシステムのリセットを行うことなしに変えることができれば大変に便利である。なぜならば最近までは国内のパーソナルコンピュータでは640ドット×400ドットという比較的解像度の低い表示画面を持ったものが広く普及していた。ところがマンマシンインターフェースの向上や表示する情報量が増えるというメリットから、より高解像度の表示画面を持ったものがでてきたが、これに対応したアプリケーションの数は少ないからである。これらの2つのモードはVRAMの大きさが異なるためメモリーの配置を変える必要があるがハードは使用中に解像度を切り換えることができてもメモリ配置やBIOSが換わるので旧いOSではこれらの切り換えを動的に行うことができず、第2のOSでの制御が必要になる。

## 【0005】

- 【発明が解決しようとする課題】ところが、新しいOSが立ち上がる前に起動されメモリに常駐しているデバイスドライバやプログラムについては新しいOSで完全に制御することはむずかしく動作モードが途中で切り換わることに対応できずに固定の動作モードのまま処理を行うため、正常動作しなくなってしまう問題が発生する。このため新しいOSを立ち上げるときはこのようなデバイスドライバ、常駐プログラムは使用できないように制限をつけるか、または複数の動作モードで正常動作できるようおののデバイスドライバ、常駐プログラムを手直しなければならなかった。

- 【0006】図9は動作モード1で第1のOSが立ち上がった直後を示している。INTnの割り込みベクタテーブル91nの内容はBIOS領域90の中の処理ルーチンA1をポイントしており、INTnによりA1が実行される。

- 【0007】図10は前記とは異なる動作モード2で第1のOSが立ち上がった直後を示しており、割り込みベクタテーブル91に動作モード1と異なる値が設定されているのが分かる。動作モード2ではINTnで処理ルーチンA2が実行される。

- 【0008】次に図11を用いて常駐プログラムの動作を説明する。パーソナルコンピュータが図9の状態で立ち上がった後、常駐プログラムbが起動されINTnをトラップする。具体的にはINTnの割り込みベク

3

ターブル91nの内容は常駐プログラムbのエントリーアドレス(処理ルーチンの先頭アドレス)bに書き換えられ、処理ルーチンA1のエントリーアドレスA1が常駐プログラムb内に格納される。以上の設定によりINTnで常駐プログラムbを介し処理ルーチンA1が実行される。

【0009】このように常駐プログラムbがメモリに常駐した後、INTnの割り込みが起こると常駐プログラムbに制御が移る。この後、常駐プログラムbの内部だけで処理が終了しINTnの呼び出し元にリターンする場合と、トラップする前のBIOSのエントリーアドレスA1に処理を戻す場合がある。常駐プログラムbが動作モード1で常駐した後動作モード2に切り換わり、トラップ前のBIOSへ処理を戻す場合に不具合が発生する。

【0010】この状態で動作モード2に切り換えた場合を図12を用いて説明する。動作モード2では割り込みベクターテーブル91及びBIOS領域90の内容が再配置され、一部を除き動作モード1との共通性はない。ここでは動作モード2のINTnの割り込みベクタアドレスは処理ルーチンA2に設定されている。ところが動作モード1で常駐した常駐プログラムbはINTnで処理ルーチンA1のエントリーアドレスをポイントしているが、処理ルーチンA1は存在せず、そのエントリーアドレスのポイントする処理ルーチンB2に制御が移り誤動作に至る。このため動作モードを切り換えるときに第2のOSでこのような常駐プログラムが内部領域に持っているトラップ前の割り込みベクターテーブルの内容を正しい値に書き直さないかぎりこれらの常駐プログラムやデバイスドライバを第2のOS以前に組み込むことはできなかった。

#### 【0011】

【課題を解決するための手段】本発明の情報処理装置は割り込みベクターテーブルを用いて割り込み処理を行う第1のOSが起動した後、前記のOS上で第2のOSが起動し、又第2のOSの上で2つ以上の動作モードを切り換えてそれぞれの動作モード上でアプリケーションプログラムを動作させることのできる情報処理装置において、第1のOSが起動した直後に任意の割り込みベクターテーブルをトラップする手段と、トラップした割り込みベクターテーブルを格納するテーブルと、第2のOS上のアプリケーションプログラムの動作モードが変更されると、前記テーブルの内容を変更後の動作モードに対応した割り込みベクターテーブルに書き換える手段と、を有することを特徴とし、動作モードの変化によって割り込みベクターテーブルの内容が変わる割り込みベクタ番号についてすべてトラップを行い内部に格納したトラップ以前の割り込みベクターテーブルの内容を第2のOS下で動作モードが変化するタイミングで正しい値に変更せざるよう第2のOSとの間でデータのやり取りを行う割り込

4

みベクターテーブル制御のための専用デバイスドライバを第1のOSが立ち上がった直後に組み込むことにより、従来からのデバイスドライバ、常駐プログラムは動作モードによる割り込みベクターテーブルの内容の変化を受けなくなり正常に動作することができる。

#### 【0012】

【実施例】図1は第1のOSと第2のOSの関係、本発明により生み出された割り込みベクターテーブルをトラップするための割り込みベクターテーブル制御用デバイスドライバ(以下デバイスドライバ)aと第2のOSとの関係を概念的に図式化したものである。第1のOS上で常駐プログラム①、常駐プログラム②、デバイスドライバ①及び第2のOSが動作し、さらに第2のOS上で各動作モードのアプリケーションが動作している。第2のOS上のアプリケーションはデバイスドライバaを介して常駐プログラム、デバイスドライバにアクセスすることになる。

【0013】第1のOSが起動された直後の割り込みベクターテーブルとBIOS領域中の処理ルーチンは従来と同様に図9で示すようになる。ここでA1、B1……は割り込みベクタであり、動作モード1と動作モード2で値が変更される。(以後動作モードは2つで説明するが、それ以上であっても本質的に同様である)

この状態にデバイスドライバaを組み込んだものが図2である。第1のOSが立ち上がった直後にデバイスドライバaが起動されると、動作モードによって変更のある割り込みベクタをすべてトラップする。割り込みベクターテーブル1上のトラップ前の割り込みベクタA1、B1……はデバイスドライバa内のテーブル3に格納され、割り込みベクターテーブル1にはデバイスドライバa内の処理ルーチンをポイントする割り込みベクタa11、a12……が設定される。INTnが実行されるとデバイスドライバa内の処理ルーチンa11の制御によりテーブル3で設定されたA1がポイントするBIOS処理ルーチンA1が実行される。

【0014】割り込みベクタのトラップ処理の手順を図5を用いて説明する。デバイスドライバaは動作モード1と動作モード2の割り込みベクターテーブルを認識しており、従って動作モード1と動作モード2での割り込みベクタを変更すればよいかを知ることができる。デバイスドライバaが起動されるとトラップが必要な割り込みベクタアドレスを順次指定する(S1)。次に指定した割り込みベクタiの内容をデバイスドライバa内のメモリ(テーブル3)に格納する(S2)。割り込みベクタiにデバイスドライバa内の処理ルーチンa1iのエントリーアドレスa1iを設定する(S3)。上記をトラップが必要なすべて割り込みベクタについて行なったことを確認し(S4)、トラップ処理を終了する。

【0015】この後常駐プログラム(デバイスドライバでもよい)bがINTnの割り込みベクタをトラップ

したのを示したのが図3である。常駐プログラムbは割り込みベクタを自身のエントリーアドレスb1に換え、デバイスドライバaの処理ルーチンa11のアドレスを自身の内部メモリ領域に格納する。INTnにより常駐プログラムb、デバイスドライバaを経てBIOS処理ルーチンA1が実行される。

【0016】この状態で動作モード2に切り換わった場合、正常動作するためにはINTxxに対するBIOSのエントリーアドレスはA1がA2に、B1がB2にといったように切り換わらなければならない。第2のOSはオペレータの指示により動作モード変更を認識し、図6で示すように、デバイスドライバaに対して動作モード1の割り込みベクタを動作モード2の割り込みベクタに変更するよう指示する。ここで第2のOSはあらかじめ動作モード1、2の割り込みベクタテーブルの値がわかっており、ある特定の割り込みまたはデバイスコントロール等によりこれらの値を順にデバイスドライバaに通知することができる。

【0017】図4は動作モード2に切り換えた後のメモリの変移の概要を示す。また図7には、前述の処理によりデバイスドライバa内のテーブル3の割り込みベクタがモード2に対応した値に切り換わった状態を示す。図4の割り込みベクタa12、a13…はデバイスドライバaの処理ルーチンの先頭アドレスにすでに変更されているのでモードが変わっても不变である。ここでINTnが発生すると割り込みベクタテーブル1内の値b1により常駐プログラムbの処理ルーチンb1に制御が移る。さらにトラップ前の処理を行うときはテーブル3に格納されているa11の値によってデバイスドライバaの処理ルーチンa11に制御が移り、処理ルーチンa11はテーブル3の値A2によりBIOS処理ルーチンA2をポイントする。よってINTnにより動作モード2の正しい処理ルーチンA2を実行することができる。

【0018】このように見てきた時、デバイスドライバaを組み込んだ後にトラップを行うデバイスドライバや常駐プログラムは動作モードの切り換えによるBIOSのエントリーアドレスの変更から切り放され、動作モードに関わりなく動作することがわかる。

【0019】なお本発明の実施例はデバイスドライバaの形式で説明したが、常駐プログラムの形式であっても、一部制約はあるが同様の効果を得ることは可能である。

#### 【0020】

【発明の効果】以上述べた如く、本発明による複数の動作モードの切り換えによる割り込みベクタテーブルの変化を制御する専用のデバイスドライバをマシンの起動時に組み込むことにより、動作モードが固定であることを前提として作成された、割り込みベクタテーブルをト

ップする常駐プログラムやデバイスドライバ（ほぼ全数がこの形態）がこれらに修正を加えることなく、動作モードの切り換えをサポートするOSで利用できるので、従来の常駐プログラムやデバイスドライバを有効に活用できるとともに前記のOSの制御が簡便になるという効果を有する。

#### 【図面の簡単な説明】

【図1】第1のOSと第2のOSと割り込みベクタテーブル制御デバイスドライバとの関係を示す概念図。

10 【図2】図9の後、割り込みベクタテーブル制御デバイスドライバaが組み込まれたときの割り込みベクタテーブルの内容の変化を示した図。

【図3】図2の後、常駐プログラムbが組み込まれたときの割り込みベクタテーブルの内容の変化を示した図。

【図4】図3の後、動作モードが切り換わり、デバイスドライバaの内部メモリに格納したトラップ時の割り込みベクタテーブルの値が変わったことを示す図。

【図5】デバイスドライバaの初期化時に、必要な割り込みベクタのトラップ処理を行なうことを示すフローチャート。

20 【図6】デバイスドライバaの動作モード変更時、第2のOSによって呼ばれる処理フロー チャート。

【図7】デバイスドライバaの内部メモリの割り込みベクタの格納領域が動作モードによって内容が書き換わったことを示した概念図。

【図8】常駐プログラムを示すメモリマップ。

【図9】動作モード1で第1のOS起動直後の割り込みベクタテーブルとBIOSとを示した図。

30 【図10】動作モード2で第1のOS起動直後の割り込みベクタテーブルとBIOSとを示した図。

【図11】図9の後、常駐プログラムbが常駐し、元の割り込みベクタテーブルの内容が常駐プログラムbの内部メモリに格納されたことを示す図。

【図12】図11の後、動作モードが2になり、BIOSの内容とそれにともなう割り込みベクタテーブルが変化することを示した図。

#### 【符号の説明】

1 : 割り込みベクタテーブル

2 : BIOS領域

40 3 : テーブル

81 : システム領域

82 : ユーザーエリア82

83 : OS領域

84 : BIOS領域

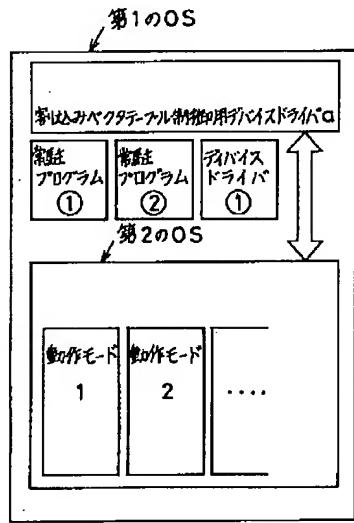
85 : 割り込みベクタテーブル

86 : アプリケーションプログラム領域

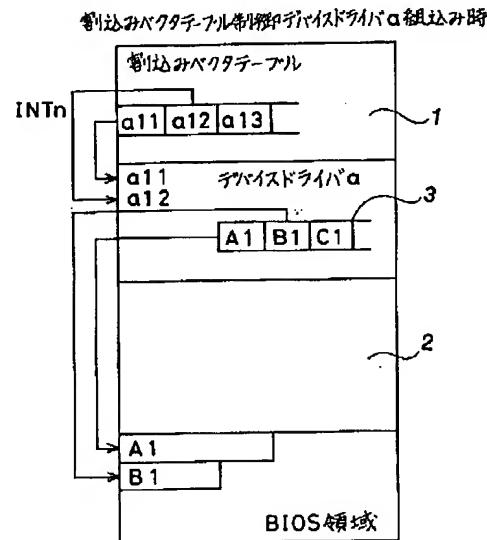
87 : 常駐プログラム領域

88 : 割り込みベクタ格納領域

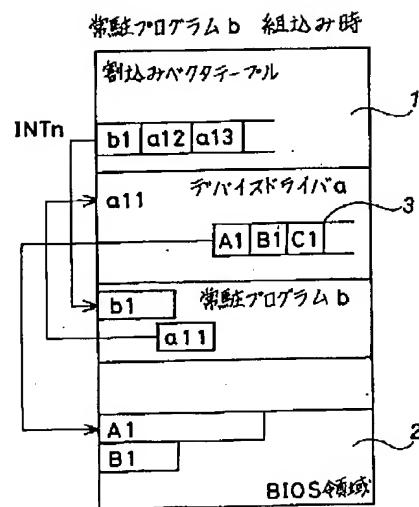
【図1】



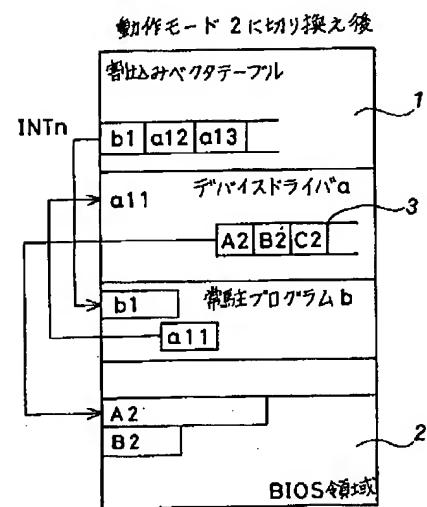
【図2】



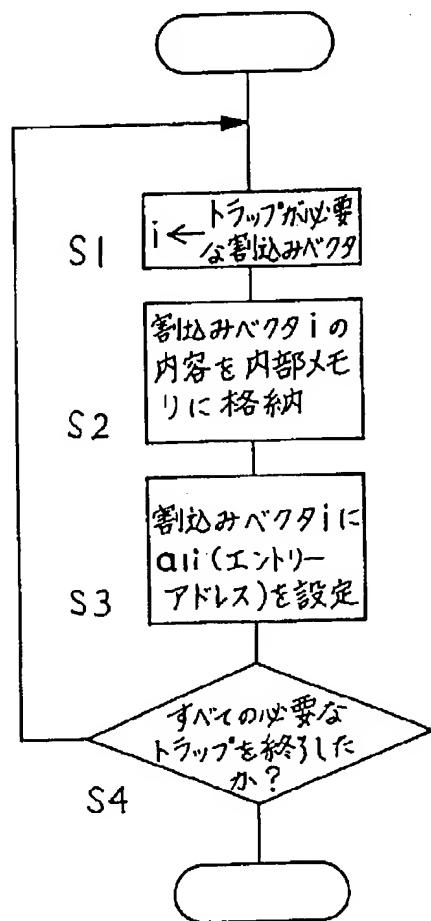
【図3】



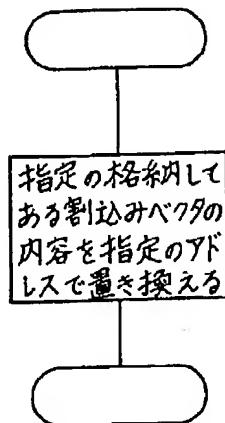
【図4】



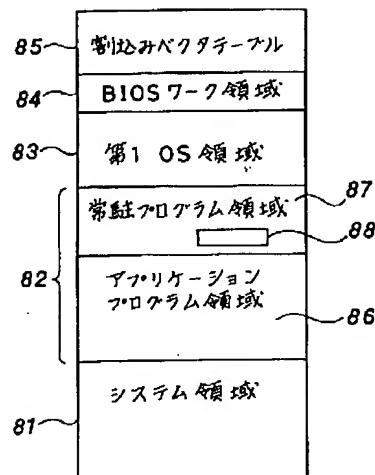
【図5】



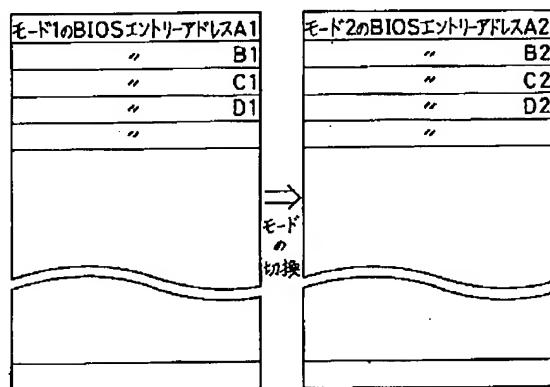
【図6】



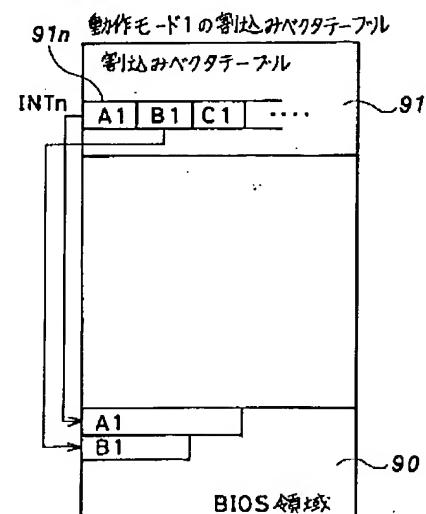
【図8】



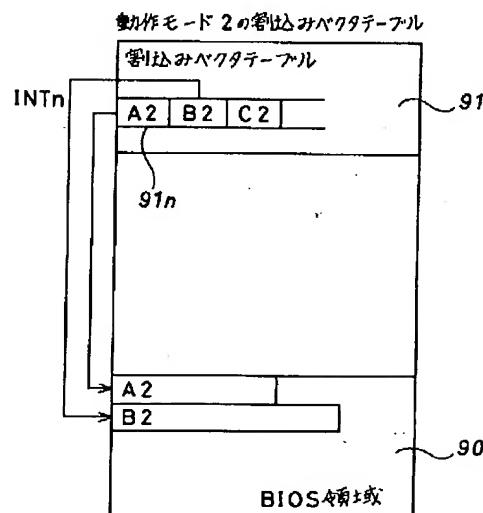
【図7】



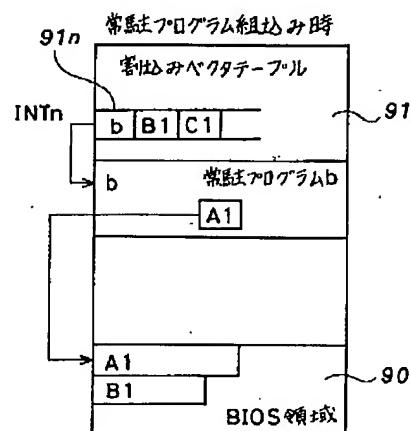
【図9】



【図10】



【図11】



【図12】

